

In the claims

1.-10. (cancelled)

11. (previously presented) A method comprising:

attempting to access a first object-oriented interface defining a method supported by an inter-related first object implemented via hardware including one or more of memory, processors, and storage devices, by a plurality of inter-related second objects, the first object-oriented interface including a password argument to limit access thereto, the first and the second objects sharing a predetermined password, the first object-oriented interface having a corresponding second object-oriented interface defining a method supported by the first object, which is required by a predetermined specification and that lacks a password argument to limit access thereto;

passing the predetermined password as the password argument of the first object-oriented interface by the second objects such that the second objects are able to invoke the method defined by the first object-oriented interface due to the predetermined password having been passed as the password argument of the first object-oriented interface; and,

attempting to access the first object-oriented interface by a third object not inter-related with the first and the second objects, the third object not privy to the predetermined password, such that the third object is unable to invoke the method defined by the first object-oriented interface, such that the first object-oriented interface is queryable by the third object but the method defined by the first object-oriented interface is uninvokable by the third object due to the third objects not sharing the predetermined password,

wherein the inter-related first object and the inter-related second objects are greater than two in number in total.

12. (cancelled)

13. (original) The method of claim 11, further comprising initially querying the first object to learn of the first object-oriented interface and the second object-oriented interface.

14. (original) The method of claim 11, further comprising:
accessing the second object-oriented interface; and,
returning a type of “not implemented” message by the inter-related first object in response to access of the second object-oriented interface.

15. (original) The method of claim 11, wherein the second object attempts to access the first object-oriented interface defining the method supported by the first object by requesting an object manager to invoke the method, such that the object manager acts as a broker between the first and the second objects.

16.-20. (cancelled)